DataGRID

PRELIMINARY EVALUATION OF REVENUE PREDICTION FUNCTIONS FOR ECONOMICALLY-EFFECTIVE FILE REPLICATION

Document identifier:	**DataGrid-02-TED-020724**
EDMS id:	**358809**
Date:	July 24, 2002
Work package:	**WP02: Data Management**
Partner(s):	**CERN and ITC-irst**
Lead Partner:	**CERN**
Document status:	**FINAL**
Author(s):	Luigi Capozza, Kurt Stockinger, Floriano Zini
File:	**eco˙model˙evaluation**

Abstract: One of the major problems in a Data Grid is the optimal distribution and replication of data files in the Grid sites, in order to improve and maintain over time a high overall throughput of Grid jobs that access files. Therefore, a Grid optimisation service [3] should be able to dynamically modify the geographic distribution of data files, triggering file deletion and replication, according to the variation over time of the sites (so called “data hot-spots”) where data is highly requested.

In this document we propose two prediction functions for evaluating the future usefulness (value) of data files. These functions can be used by Grid sites to make informed decisions about whether or not to replicate files locally. Both functions use for their prediction logs of file requests that jobs have submitted to the site but assume different statistical models for the historic data.

We have performed some preliminary tests on the accuracy of these prediction functions using randomly generated simulated file access patterns. We have compared the predicted values with the simulated ones. It turns out that, over the performed tests, the two functions behave similarly and predict the simulated values reasonably well.

CONTENTS

1. INTRODUCTION

In [2] we propose an approach, based on an economic model, for optimising file replication in a Data Grid. In short, in the model there are two main classes of actors. Computing Elements (CEs) have the goal of making data file available for access to Grid jobs on the site where they are executing. CEs try to purchase the cheapest replicas of needed files by interacting, via an auction protocol, with Storage Brokers (SBs) located in the Grid sites. SBs have the goal of maximising revenues they can obtain by selling files to CEs or other SBs. In the economic model we make the assumption that the usefulness of a file is proportional to the revenue a SB can obtain from it. SBs have to decide whether replicating a file to the local site is a worthwhile investment. Since Grid sites have finite storage space, this could also result in deleting other files.

In order to make a replication (and deletion) decision, SBs may use various strategies. In particular, in [2] we propose the use of a prediction function that estimates the future revenue of a stored file based on the past revenue of this file and of files with similar contents. In this paper we define two types of such prediction function and present some experimental evaluation we have performed on them.

The document is organised as follows. Section 2. introduces the model of file requests that we assume for the definition of the prediction functions, which are described in Section 3.. Section 4. presents the preliminary results we have obtained using simulated file access pattern and finally Section 5. and 6. briefly discuss replication strategies based on the prediction functions and how these can be tested in a Data Grid simulator we are currently developing [4].

2. FILE REQUESTS AS A RANDOM WALK THE SPACE OF FILE IDENTIFIERS

During the execution of Grid jobs a SB receives a sequence of file requests, either from CEs or other SBs that have been requested a file that they do not have. We make the following two assumptions on this sequence.

Request arrivals with exponential distribution. As for many similar phenomena that have been studied in queueing theory, we can assume time intervals between two successive file requests being independent random variables with exponential probability distribution of parameter τ.

Therefore, after each file request (e.g., arrived at time t_0), the probability $\phi(t)$ that a new requests arrives within the time interval $t - t_0$ is given by

$$\phi(t) = 1 - e^{-\tau(t-t_0)} \tag{1}$$

where the decay rate τ represents the average number of requests in a time unit.

Sequential correlation between file requests. We assume that SBs receives over time file requests that are *sequentially correlated*. This means that, given a file request, there is a great probability of successive requests for files with similar content.

We can model content similarity by first defining the *file space* $\{F\}$ as the set of all the potentially requested files and the *file-ID space* $\{f\}$ as a set, with the same cardinality as $\{F\}$, of file identifiers. File identifiers are assumed to be integer positive numbers. We can then define content similarity as a mapping M between $\{f\}$ and $\{F\}$ as follows. Assuming two file-IDs f_1 and f_2, the closer they are, i.e. the smaller the difference $|f_2 - f_1|$, the higher is the expected relation between the contents of the corresponding files F_1 and F_2[1].

[1] The definition of file content similarity in a Data Grid is still an open problem. Here we just give a definition of it that can be used in the prediction functions we have defined.

Given the assumptions above, we can represent the history of file requests as a *random-walk* in the space $\{f\}$ of file identifiers. A random-walk consists of a sequence $\langle f_i, i \geq 0\rangle$ of identifiers and is obtained starting from an initial identifier f_0 and adding a sequence $\langle s_i, i > 0\rangle$ of *random walk steps*, each of which leads from f_{i-1} to f_i, i.e. $s_i = f_i - f_{i-1}, i > 0$.

Each generic step s is an independent random variable which may assume values within the interval $[-S, +S]$ where $2S$ is the maximal difference between the identifiers of two successively requested files. This way, we can model a symmetric random walk with variable step size.

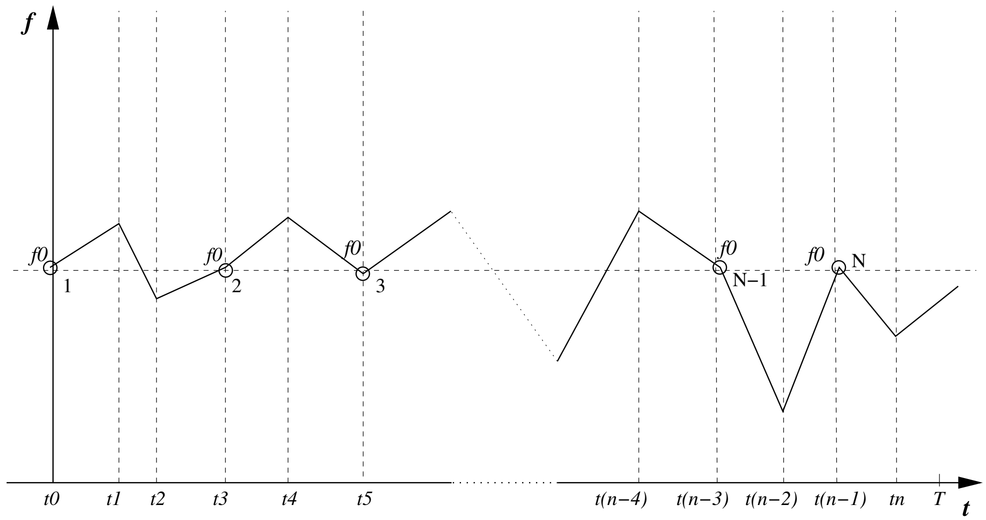

Figure 1: File access history as a random-walk in space $\{f\}$ of file identifiers.

Figure 1 shows an example of random walk. Arrival times for file requests and corresponding file identifiers are shown along the t and f axis respectively. In the example, the walk starts from file f_0 (requested at time t_0) and covers n file requests over a time T. Among the n requests, file f_0 is requested N times.

3. Prediction of Future File Revenues

In [2] we define a function that calculates the revenue that a SB obtains over a (future) time period T by selling a file F, a request for which is received at time t_k. The value of the file is obtained summing up the incomes the SB receives for F over T. Here we consider a simplified form of this function, defined as

$$V(f,k,n) = \sum_{i=k}^{k+n} \delta(f, f_i) \tag{2}$$

The function $V(f,k,n)$ calculates the revenue for the file F corresponding to the identifier f starting from time t_k and taking into account the next n file requests[2]. Moreover, we consider unitary file prices and thus within the sum there is only the Kronecker delta function[3] between identifiers f and f_i. Equation (2) states that $V(f,k,n)$ is given by the number of times file f will be requested during the next n requests, starting from request k.

[2]Note that the domain of $V(f,k,n)$ is the set $\{f\}$ of file identifiers. In the following we will sometimes use the term *file* when it would be more correct to use the term *file identifier*. However, it is clear that we will always refer to the file corresponding to the file identifier.

[3]Kronecker delta is defined as:

$$\delta(f_i, f_j) = \begin{cases} 0 & \text{if} \quad f_i \neq f_j \\ 1 & \text{if} \quad f_i = f_j \end{cases}$$

In order to estimate $V(f,k,n)$ we have to predict how many times the file f will be requested over the next period of time considered. We use a simplified version $E[V(f,k,n),r]$ of the prediction function defined in [2]. The index r indicates that estimation is made on the basis of the r file requests received by the SB before the request for file f.

3.1. Defining the Prediction Function

Our goal is now to obtain an analytical form for $E[V(f,k,n),r]$. Based on the latest r observed requests, $E[V(f,k,n),r]$ should provide a good estimation for the number N of requests for file f that are going to be received among the next n file requests.

As stated in Section 2., file access history can be represented as a random-walk in the space $\{f\}$ of file identifiers, consisting of a sequence $\langle f_i, i \geq 0\rangle$ of identifiers starting from an identifier f_0. In order to perform calculations and get to an explicit form for the evaluation function $E[V(f,k,n),r]$ we choose a particular shape for the probability distribution of a generic s, which represents the step between two successive identifiers in the random walk.

3.1.1. Prediction Function Based on a Binomial Distribution

Here, we assume that a generic step s in the random-walk is a discrete random variable with binomial distribution[4].

We can thus imagine for each s a distribution like in Figure 2. Such a function is obtained by using a binomial distribution, making it symmetric and centring it on zero. This allows us to consider a symmetric random-walk with integer random step size, always belonging to the interval $[-S,S]$. Moreover, we can model the fact that once a file has been requested, it is the most probable requested file in the next step of the random walk.

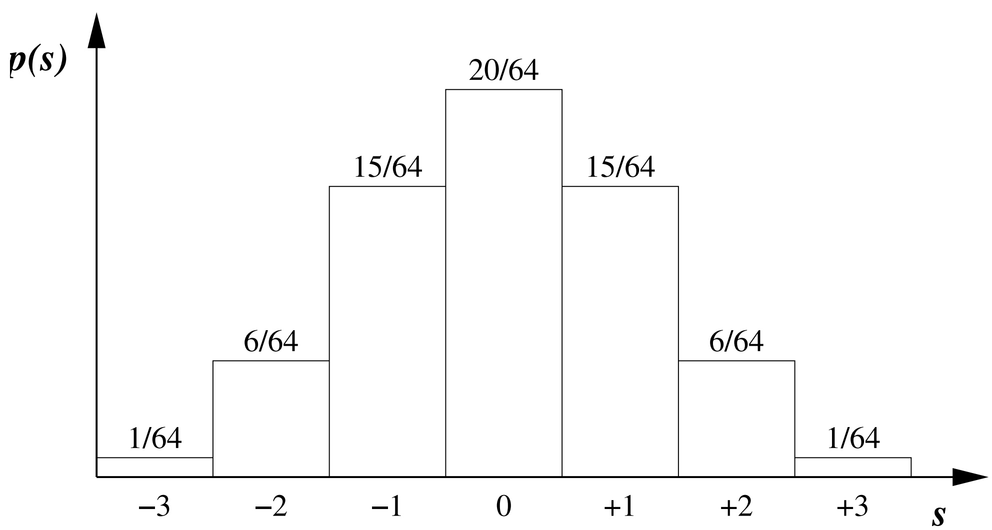

Figure 2: Binomial distribution centred on zero and with $S = 3$.

[4]The binomial (or bernoullian) distribution returns the probability $p(k)$ to obtain k successes performing n independent trials of a certain test when the probability of success of each single trial is q. It is given by:

$$p(k) = \binom{n}{k} q^k (1-q)^{n-k} \qquad 0 \leq k \leq n$$

In case $q = 1/2$ the distribution is symmetrical around its central value $n/2$.

The definition of the distribution $p(s)$, its mean value $\bar{s}$ and its standard deviation σ_s are given by

$$p(s) = \frac{1}{2^{2S}} \binom{2S}{s+S}, \qquad \bar{s} = 0, \qquad \sigma_s = \sqrt{\frac{S}{2}} \tag{3}$$

For example, with reference to Figure 2, the probability of $s = -1$ is given by

$$p(-1) = \frac{1}{2^{2*3}} \binom{2*3}{-1+3} = \frac{15}{64} \tag{4}$$

Each identifier f_i represents the file requested at the i-th step in the random-walk and is obtained starting from the initial identifier f_0 and following the path through the first i steps of $\langle s_i \rangle$. Thus we can write

$$f_i = f_0 + \sum_{j=1}^{i} s_j \tag{5}$$

Since each s_j is an independent random variable with symmetric binomial distribution, each generic f is also a random variable with the same distribution. Its mean value and standard deviation are given by

$$\bar{f} = f_0, \qquad \sigma_f = \sqrt{\frac{iS}{2}} \tag{6}$$

We can now find a simple expression for the probability distribution of f. The value s in (3) can be regarded as the sum of $2S$ values chosen independently, with a probability $q = 1/2$, within the set $\{-1/2, +1/2\}$[5]. The sum at right hand in (5) may therefore be interpreted as the sum of $2iS$ values chosen (independently and with probability $p = 1/2$) from $\{-1/2, +1/2\}$. That sum represents the net movement from initial the position f_0 to the final position f_i.

Setting the mean value of f equal to $\bar{f}$[6] instead of f_0, the probability $p(f)$ of receiving a request for file f at step i is given by

$$p_i(f) = \frac{1}{2^{2iS}} \binom{2iS}{f - \bar{f} + iS}, \qquad |f - \bar{f}| \leq iS \tag{7}$$

Let us imagine now to have a number R of sequences of file requests each containing n requests and each starting from the same position $\bar{f}$. If by $r_i(f)$ we mean the number of times file f has been requested at step i through the R sequences, then the ratio

$$\frac{r_i(f)}{R} \tag{8}$$

will approximate to $p_i(f)$ for increasing R.

Summing up all terms $r_i(f)$ for i going from 1 to n, i.e., $\sum_{i=1}^{n} r_i(f)$, we obtain the total number of times that file f has been requested during the R sequences (a total of nR requests). Dividing this sum by R, we get the average number of requests for f in a single sequence. This value is equal to the sum of all terms like (8) (for i going from 1 to n) and represents the *most probable number of times file f will be requested during the next n requests*, which is exactly the evaluation $E[V(f,k,n),r]$ we are looking for. Assuming that $p_i(f)$ is the best estimation of (8), we can write

$$E[V(f,k,n),r] = \sum_{i=1}^{n} p_i(f) \tag{9}$$

Note that the dependence on r, i.e. the number of requests received in the last period, for the right hand term is hidden within the definition of parameters $\bar{f}$ and S appearing in (7) as we will see below.

We finally define the parameters n, $\bar{f}$ and S.

[5] Note that the sum of an even number of such values always leads to an integer value ranging from $-S$ to $+S$.

[6] Later it will be discussed how to fix this value.

n First of all we have to fix a time interval T' in the past that serves as the basis for our estimation. Then T be the future interval for which we intend to do the prediction. So, r is the number of requests received in the last period of length T'. By assuming that the arrival rate of requests will stay the same, n is obtained by:

$$n = r\frac{T}{T'} \tag{10}$$

Obviously if $T = T'$ then $n = r$.

S We want to give an estimation of the width S on the basis of the past. Suppose going backward in time from t_0 to t_{-r}. Doing a reverse random-walk and considering Equations (6) we can say that for each j going from 1 to r, variable f_{-j} should have a variance given by $jS/2$. The best estimation we have for this variance is $(f_0 - f_{-j})^2$. Calculating a weighted average on these quantities we obtain an expression for S:

$$S = \frac{2}{r}\sum_{j=1}^{r}\frac{(f_0 - f_{-j})^2}{j} \tag{11}$$

$\bar{f}$ The simplest way to fix a central value for our sequence of distributions on the basis of previous requests is to do a weighted average among last r files requested where the weights decrease going toward the past. That is, the more recently a file has been requested, the more important it is in calculating $\bar{f}$.

It could be possible to not consider the history at all in choosing $\bar{f}$. In this case one can simply decide to set all the weights equal to zero and thus $\bar{f} = f_0$.

3.1.2. Prediction Function Based on a Normal Distribution

If one decides to adopt the last perspective mentioned in Section 3.1. and thus not to take into account the history to fix the starting point $\bar{f}$, then all the functions $p_i(f)$ are centred on $\bar{f} = f_0$. This will always lead to decide in favour of replication, at least while file prices are supposed to be unitary. This happens because in such a case only the estimated number of times a file will be requested in the future is important. This is provided by the function in Equation (9) which always has its maximum in $\bar{f}$, being (7) the form of $p_i(f)$.

In order to address this problem, one might try to extrapolate the moving trend (in the space of file identifiers) of future file accesses from previous ones. This can be done by:

- calculating the mean displacement $\bar{s}$ in the latest requests, i.e. the mean value of the latest steps in the random walk.
- supposing this mean displacement will stay the same in the next period;
- centring successive $p_i(f)$, where i refers to the i-th step in the random walk, not on f_0 but on $f_0 + i\bar{s}$, to follow the trend.

The first raising problem with such a procedure is that we need a distribution for f_i (and for s_i as well) which can be calculated even for non-integer values. The simplest choice is the Gaussian distribution with the same standard deviation $\sqrt{iS/2}$.

The second problem, consequently, is that using a continuous distribution we cannot calculate the probability for a certain value of the variable f but we must consider an interval $[f-d, f+d]$, which gives approximately the probability of receiving a request for files within the range of width $2d$ and centre in f. So, considering the probability distribution $\tilde{p}_i(f,d)$ defined by

$$\tilde{p}_i(f,d) = \frac{1}{\sigma_i\sqrt{2\pi}}\int_{f-d}^{f+d} e^{-(x-i\bar{s})^2/2\sigma_i^2}\,dx, \tag{12}$$

where σ_i is σ_{f_i} as in Equation (6), we have, in obvious notation, a new prediction function

$$\tilde{E}[\tilde{V}(f,d,n),r] = \sum_{i=1}^{n} \tilde{p}_i(f,d). \tag{13}$$

To do a comparison between the two prediction functions, even the value that E should predict must be changed. Thus it will not be anymore the number of times file f will be requested during the next n requests but the number of requests for files within the range of width $2d$ and centre in f (in our notation $\tilde{V}(f,d,n)$).

4. Preliminary Tests and Results

In this section we present preliminary tests for the prediction functions defined in (9) and (13). In order to perform these tests, we first generated some sequences of file requests and thus obtained some random-walks like the one in Figure 3[7]. The figure shows a random-walk with about 1000 file requests for files whose identifier is in the range $[20, 100]$. In the following, we refer to these file requests as either *real* or *simulated*.

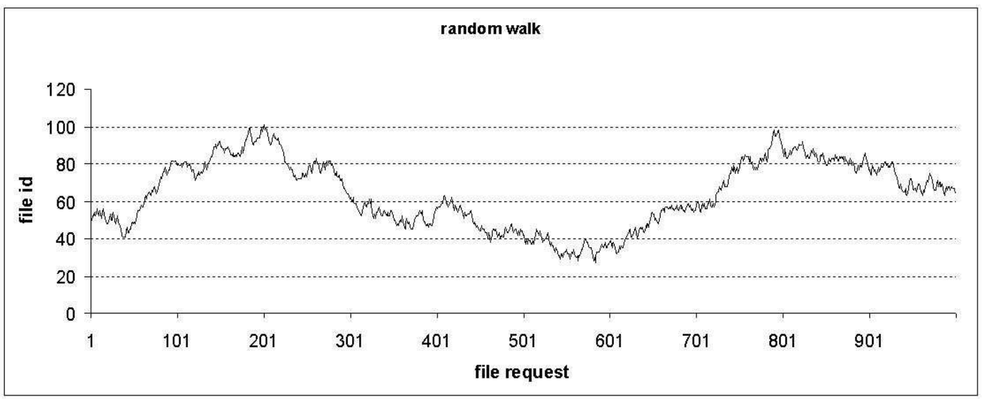

Figure 3: Random walk describing simulated access pattern.

Given a certain position within the sequence of file requests, we calculated $E[\tilde{V}(f,d,n),r]$ and $\tilde{E}[\tilde{V}(f,d,n),r]$, which give the prediction of the future number of requests for file f. In addition, we calculated the real file value $\tilde{V}_n(f,d,n)$ by counting the number of times file f was requested from that position on. Iterating over a segment of the sequence we obtained a first qualitative evaluation of the performance of these functions.

We performed two types of evaluation tests:

1. calculation of the average of the squared differences between predicted and real values over a segment;

2. graphical representation of simulated and estimated values over a segment and comparison between these graphs.

[7]It could seem not very correct to test a prediction function, formulated conjecturing the access pattern to be a random walk, by the use of a random walk test-set. In fact, it is a fairly lower constrain to suppose a random walk with general step distribution rather than one with fixed step distribution, which is the case of the reasoning that led to our functions. This seems to be a simple way to produce a set of data reflecting the characteristics the file accesses are supposed to have.

The first type of test did not give significant results. We performed the test using some random-walks but we could not find a prediction function for which the statistical measure was better than the other one.

The result of the second type of test performed over a sequence of simulated file requests is shown in Figure 4. The figure compares the two prediction functions $E[\tilde{V}(f,d,n),r]$ and $\tilde{E}[\tilde{V}(f,d,n),r]$ (which are based on the assumption that the file access pattern follows a binomial and a gaussian distribution, respectively) and verify them against the real values obtained from the simulated file requests.

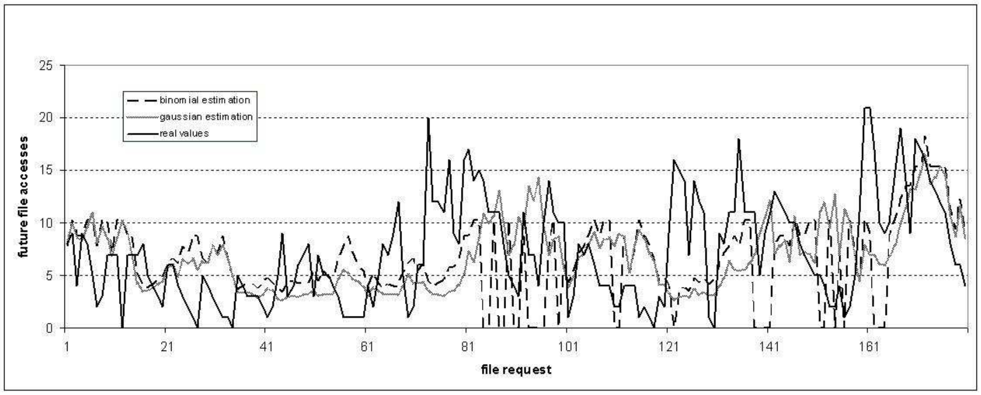

Figure 4: Progress of predicted values and real values over a sequence of file requests.

This graph shows that the two functions gives similar results and predict quite accurately the real values. We also compared the two prediction functions using other sequences of simulated file requests and came to the same conclusion. However, since we did not perform a sufficient number of experiments, we cannot come to the conclusion that one prediction function is superior over the other one or that the proposed prediction functions reflect accurately the actual pattern of file request.

5. REPLICATION STRATEGIES

When a SB is given the possibility to locally replicate a files, it can use various strategies to decide whether or not this is the case. A winning strategy should be able to answer the question:

> Given a finite storage space and some concrete access patterns, which files should be kept and which ones should be deleted in order to optimise data locality and thus overall system performance?

An example of a replication reasoning based on the prediction functions previously defined could be the following. When a SB is given the possibility to buy a file f_0, it calculates $E[V(f_0,k,n),r]$ and $E[V(f,k,n),r]$[8] for fixed r and n and for each $f \in F$, where F is the set of files that are stored in the corresponding SE. If there is an f within F which verifies $E[V(f,k,n),r] < E[V(f_0,k,n),r]$ then f is replaced by f_0.

Predicting the future value of a single file, as done in Section 4. is far beyond the actual scope of evaluating a winning replication strategy. We are currently analysing which replication strategies lead to optimise Grid performance. This is done using a Grid simulator.

[8] We could also use $\tilde{E}[\tilde{V}(f,n,d),r]$.

6. Testing Replication Strategies with a Grid Simulator

Part of our research is currently to test the effectiveness of various replication strategies by using a Grid simulator [4, 1]. This simulator behaves like a real multi-site Grid where a certain number of jobs are scheduled to different Grid sites. These jobs request files querying (directly or indirectly) a number of SBs, that in turn may or may not have the requested files. When a file is finally found, the involved SBs make a decision about local replication of the file.

In order to evaluate different replication strategies we will measure some relevant parameters, still to be defined, that give a sight of the overall performance of the simulated Grid. By running the simulation for a certain (*big*) number of Grid jobs using various replication strategies we will be able to make a comparison and thus to choose the most effective strategy.

One parameter that could be relevant for our evaluation is the total gain of SBs in the Grid at the end of the simulation. This could be regarded as a measure of how good the replication policies have been. This because of the following speculations, the verification of which is part of our future research:

- If the total gain of SBs is high this could mean that the total cost for transporting files (that is charged to SBs) from one site to another is low. This, in turn, means that access to files has been optimised.
- If the total gain is high this could also mean that high prices are paid for files. According to the payment mechanism proposed in [2] this means that replicas were created only on strategic sites without over-replication. In such a case, the number of replication processes has been minimised.

7. Acknowledgements

We would like to thank our colleagues within WP2 for their feedback. Special thanks go to William Bell, Diana Bosio, David Cameron, Ruben Carvajal Schiaffino and Paul Millar for their fruitful comments.

References

[1] W.H. Bell, D.G. Cameron, L. Capozza, A.P. Millar, K. Stockinger, and F. Zini. Simulation of Dynamic Grid Replication Strategies in OptorSim. Technical Report DataGrid-02-TED-3613, EU-DataGRID Project, June 2002.

[2] M. Carman, F. Zini, L. Serafini, and K. Stockinger. Towards an Economy-Based optimisation of File Access and Replication on a Data Grid. In H.E. Bal, K. Lör, and A. Reinefeld, editors, *Proc. of 2nd IEEE/ACM Int. Symposium on Cluster Computing and the Grid (CCGrid'2002)*, pages 340–345, Berlin, Germany, May 2002. IEEE Computer Society. Also appears as Technical Report 0112-04, ITC-irst, September 2001.

[3] WP2 Data Management Query Optimisation Task. Design of a Replica Optimisation Framework. http://edms.cern.ch/document/337977/.

[4] WP2 Data Management Query Optimisation Task. Optor - Query Optimisation Simulation. http://grid-data-management.web.cern.ch/grid-data-management/optimisation/optor/.